International Journal of Computer Networking,
Wireless and Mobile Communications (IJCNWMC)
ISSN(P): 2250-1568; ISSN(E): 2278-9448
Vol. 4, Issue 2, Apr 2014, 159-172
© TJPRC Pvt. Ltd.

DESIGN AND IMPLEMENTATION AN E-VOTING SYSTEM BASED ON SOPHISTICATED TECHNOLOGIES OF WIRELESS NETWORKS AND VISUAL PROGRAMMING LANGUAGES

MAHMOOD ZAKI ABDULLAH

Department of Computer & Software Engineering, The University of Al Mustansiriyah, Baghdad, Iraq

ABSTRACT

Electronic voting systems are the latest voting systems currently approved in many countries of the developed world, which has struggled to transform their traditional voting systems to electronic systems based on computerized ways and modern aspects of information technology. This paper gives a suitable design of a hardware system for a voting system that is electronically sophisticated by use of computerized mechanism that depends on computers associate with each other through a reliable wireless network using modern routers to guide and transfer packets. A software package, has also been building by use of a modern Visual language, like a Visual Basic.Net language, in order to achieve and test the hardware and software validity of this system. The hardware design of this system was tested practically by connecting a group of computers through a wireless network, the quality of all the signals transferred in the network were checked through the use of special software. The proposal software package was also implemented and examined practically, the results obtained were very good and without any errors, according to these results, this proposal system can be considered as a suitable programmatic design for modern electronic voting systems.

KEYWORDS: E-Voting, Wireless Network, Visual Programming

INTRODUCTION

E-voting is one of the important use of the modern information technologies, many countries now a day the E-Voting in their election systems, supporting group decisions has become an important topic in the field of computer applications, and electronic voting (e-voting) has received a great deal of attention in recent decades. People can use modern digital devices such as PCs, PDAs, cell phones, or laptops and networks such as Internet, Intranet, wireless networks, or Ad hoc networks to make group decisions electronically.[1]

The briefly description of the essential criteria is that a secure e-voting system should satisfy the following requirements:

Computer Networks

A voting system must consists of several computer devices that were connected via networked, may be used internetworking devices such as hub, switches, routers, and even gateways to establish this networked connections.

Server

The intrinsic and important device that the E-voting system must contain is the server, this server can store the necessary data base for the voter, and the other data.

Software Package

Before beginning the vote, each voter must pass a series of voter authentication processes. When the voter passes the authentication process, he/she is permitted to join the vote. In other words, only eligible voters are allowed to vote, to do that a suitable secured software package can be applied.

RELATED WORKS

In this section we review some related works regarding the basic concepts of e-voting, and cryptographic techniques used in our work.

Chaum [8] proposed the first electronic voting method that enables voters to electronically cast his/her ballot over insecure networks. Later, there were persistent efforts made by researchers to increase security and to make the applications of e-voting more comprehensive, several e-voting protocols have been proposed in the literature [1, 2, 3, 4, 5, 6, 7].

Chen. [9], Chang and Lee [7], introduced a public proxy server that has the advantage of increased anonymity during the voting phase. The voter sends his/her vote to the voting center through a trusted proxy server, and the original network addressof voter can be replaced by a proxy address [1, 9].

Ghassan Z., Qadah, and Rani Taha [10], designed and implementated a special type of electronic voting systems, the remote on-line voting system, suitable for university setting where students can cast their votes anytime, anywhere and using fixed and mobile electronic devices including personal computers, personal digital assistants and smart and regular phones. They suggested a new way to avoid web content replication for each of the connecting devices, the implemented system separates the data content from its presentation form. The separation is achieved by using modern technologies such as the extensible mark up language to represent the web data content and the extensible style language transformation style sheets to customize the presentation of such content on different connecting devices [10].

CONVENTIONAL E-VOTING SYSTEM

Prior to e-voting system design, a comprehensive and detailed set of requirements must be developed. These requirements include three main types, namely, generic, system specific and election-specific requirements [10].

Generic Requirements [10]

The generic requirements are those requirements that apply to any voting system. These requirements, as shown in Figure 1, include:

- **Authenticity:** only eligible voters can cast their votes;
- **Integrity/Accuracy:** once a voter cast a vote, no alternation to this vote is permitted. Moreover, all valid votes must be counted, whereas all invalid ones must not be counted;
- **Privacy:** after casting a vote, no one should be able to link the voter to this vote;
- **Security:** throughout the voting process, a vote can't be tampered with or viewed by anyone;
- **Democracy:** All eligible voters must be able to vote, one person–one vote and no one can vote more than once or vote for others.

System-Specific Requirements [10]

The system-specific requirements, on the other hand, are those requirements that are specific to on-line electronic-voting systems. The system-specific requirements include:

- **Multi-User:** A number of voters can vote simultaneously;
- **Multi-Elections:** A number of elections can be running simultaneously;
- **Accessibility:** The system can be accessed by voters at anytime, from any-location using the Internet/Intranet, PDAs, and/or mobile phones;
- **Availability:** The system must have high-availability during an election campaign.

Election-Specific Requirements [10]

The election-specific requirements are those requirements that are specific to a given election. For example, the election specific requirements for student council election include the following items:

- A voter must be registered as a full time student;
- A candidate must be registered as a full time student;
- A candidate must have completed at least two semesters at the University and must maintain a GPA of at least 2.5 (out of 4.0) at the time of nomination.
- A candidate must not serve more than two consecutive years. A candidate is also a voter. An election winner must be one of the candidates for that election.

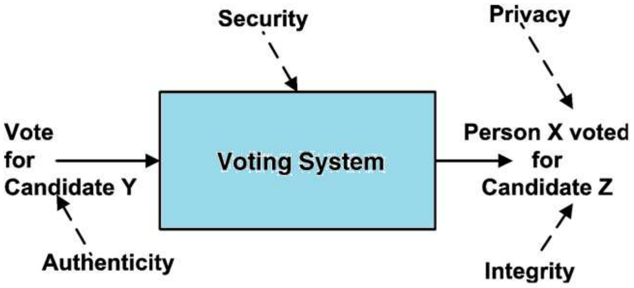


Figure 1: A Generic Set of Requirements [10]

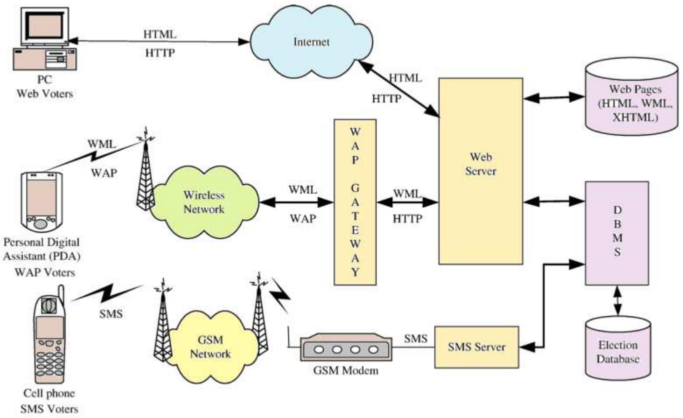


Figure 2: The Organization of a Generic Remote E-Voting System [10]

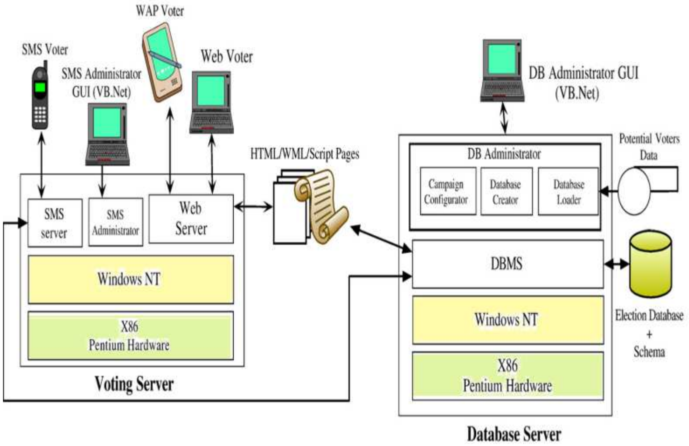


Figure 3: The Components of the Implemented E-Voting System [10]

A generic e-voting system, as shown in Figure 2, consists of a number of components, the election database stores the data elements representing elections, candidates and voters as well as the schema that describes these data elements and the associated constrains. These constrains implement some of the election-specific requirements presented above. The election database is managed by a modern database management system (DBMS), such as Oracle, Sybase, or MS-SQL [10], the developed system, as shown in Figure 3, uses two hardware servers, the database and the voting ones. The database server runs a modern Database Management System (DBMS). In addition, it supports a number of software tools [10].

HARDWARE DESIGN OF A PROPOSAL E-VOTING SYSTEM

From the previous section, we found that the conventional generic e-voting system that was shown in figure 3 consisted of computers, servers, modems, and using many techniques and type of networks like Internet, and free GSM for mobile, this system gave the variety of sending the voted signal from the voter (computer or mobile devices) to the main or (chairman) computer or coordinator computer. From these facts we designed a proposal e-voting system, this proposal e-voting system differed from the conventional e-voting system in the way of using networking connections, the types of internetworking devices, the type and position of servers, in our proposal design we connected the main and coordinator computers with special server called main server, this connection was done by using wireless received wired port router, while the voters PCs. and mobiles were connected to a special server called voters server, this connection was done by using wireless router, all these networks were connected together by using a Wi Fi signal, as shown in figure 4,

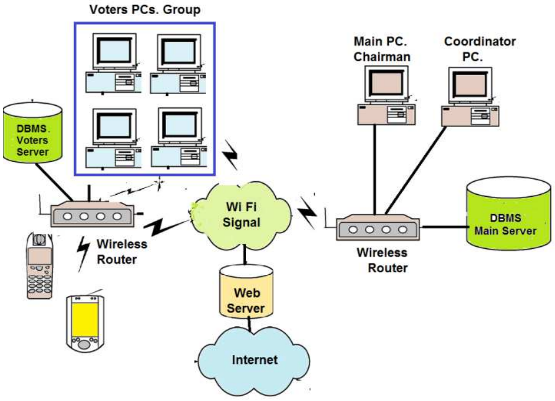


Figure 4: Proposal E-Voting System

Before we test our proposal design of e-voting system and implement it practically, we took many scenarios of networked system with various internetworking devices, designed and test them by using the packet tracer v.5 simulator, as shown below.

Scenario - 1

In this scenario we connected (15) computers (Hosts PCs.) to main computer (chairman pc.) and server through a switch, each host could send its vote to main PC. and could take any data that might be need from the data base's server.

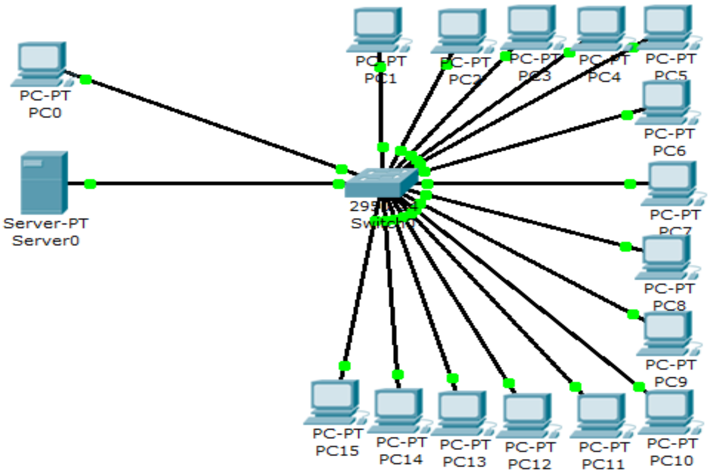


Figure 5: Scenario-1 Wired Networked E-Voting System

Scenario - 2

In this scenario we connected (15)computers (Hosts PCs.)to main computer (Boss PC.) and server through a wireless router, we connected the main pc. and server through a wired networked to a wireless router, each host could send its vote to main pc. and could retrieve and store data to the server.

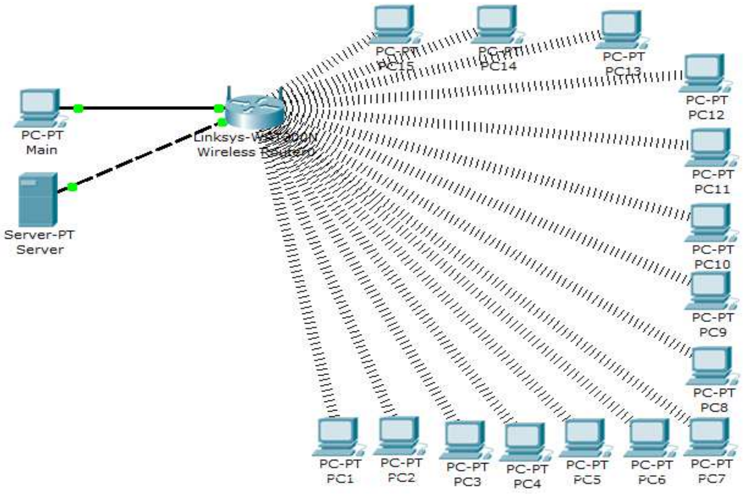


Figure 6: Scenario-2 Wireless Networked E-Voting System

SOFTWARE DESIGN OF A PROPOSAL E-VOTING SYSTEM

The software design of the proposal e-voting system can be done and programmed for two types of computers; the first type was the main computer, while the second type was for voter computers.

Main Computer Program

The proposal software package of the main computer can be designed and programmed according to the following steps, these steps described how the main computer (the computer of the Chairman of the Board) can notice and score the votes of the members (the computers of voters) for certain law.

- After reading the law or paragraph which needs to vote for, voting begins by pressing ”begin voting”.
- Given limited duration time for vote (voting period), then the results are displayed and find out if the law was approved or not approved.
- In the event of an equality vote, the opposition and the approval of the law is re-discussion about the law and vote again.
- By knowing the number of voters, can find out if a full quorum, if no vote is canceled.
- Through the main computer, know a certain choice for voters by writing his name.

Voter Program

The voter program can be designed to cover the following steps:

- After reading the law and the start of the voting process, voters are allowed to choose(Yes, No, Idle) during a certain period.
- Voters are not allowing him to switch his vote during the time more than once.
- After the period, voting is suspended in order to collect the results of voting.

The Program Codes for a Main Computer

The source codes of the main computer program, were written to cover many functions, like (Begin vote, The Results), these codes were written in visual basic.net program, figures 7 and 8 show the screenshot for the main and code for load.

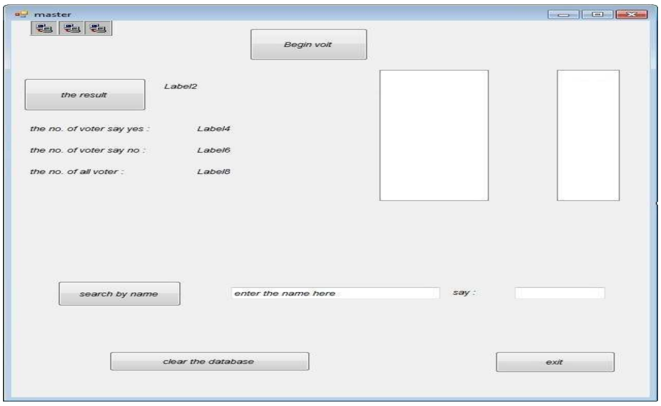
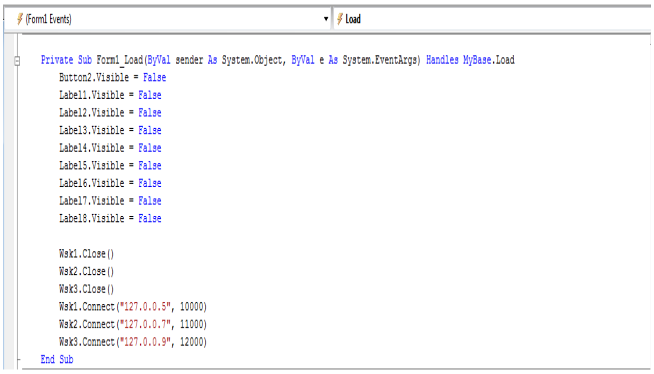
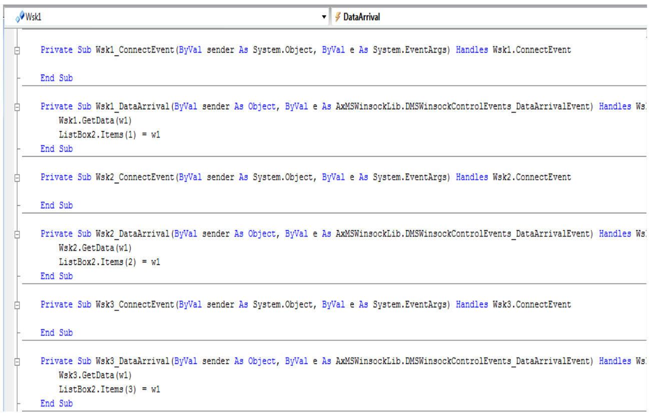


Figure 7: Screenshot of Main Computer Program



```
Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
    Button2.Visible = False
    Label1.Visible = False
    Label2.Visible = False
    Label3.Visible = False
    Label4.Visible = False
    Label5.Visible = False
    Label6.Visible = False
    Label7.Visible = False
    Label8.Visible = False

    Wsk1.Close()
    Wsk2.Close()
    Wsk3.Close()
    Wsk1.Connect("127.0.0.5", 10000)
    Wsk2.Connect("127.0.0.7", 11000)
    Wsk3.Connect("127.0.0.9", 12000)
End Sub
```

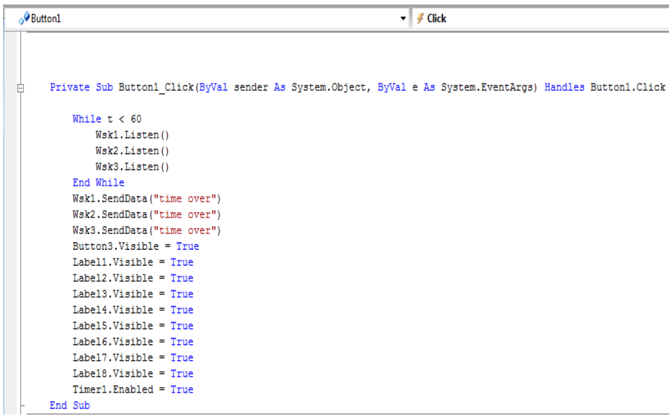
Figure 8: Screenshot of Code Load



```
Private Sub Wsk1_ConnectEvent(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Wsk1.ConnectEvent

End Sub

Private Sub Wsk1_DataArrival(ByVal sender As Object, ByVal e As AxMSWinsockLib.DMSWinsockControlEvents_DataArrivalEvent) Handles Ws
    Wsk1.GetData(w1)
    ListBox2.Items(1) = w1
End Sub

Private Sub Wsk2_ConnectEvent(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Wsk2.ConnectEvent

End Sub

Private Sub Wsk2_DataArrival(ByVal sender As Object, ByVal e As AxMSWinsockLib.DMSWinsockControlEvents_DataArrivalEvent) Handles Ws
    Wsk2.GetData(w1)
    ListBox2.Items(2) = w1
End Sub

Private Sub Wsk3_ConnectEvent(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Wsk3.ConnectEvent

End Sub

Private Sub Wsk3_DataArrival(ByVal sender As Object, ByVal e As AxMSWinsockLib.DMSWinsockControlEvents_DataArrivalEvent) Handles Ws
    Wsk3.GetData(w1)
    ListBox2.Items(3) = w1
End Sub
```

Figure 9: Screenshot of Source Codes for Main Program

While the source code for ‘Begin Vote” key function can be represented by figure 10.



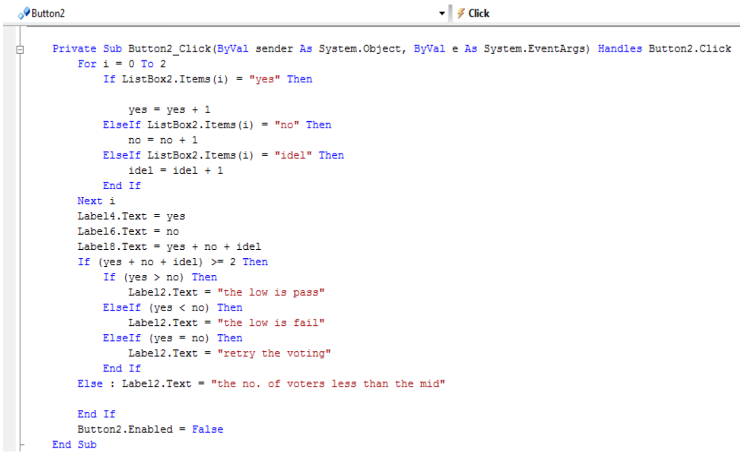
```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click

    While t < 60
        Wsk1.Listen()
        Wsk2.Listen()
        Wsk3.Listen()
    End While
    Wsk1.SendData("time over")
    Wsk2.SendData("time over")
    Wsk3.SendData("time over")
    Button3.Visible = True
    Label1.Visible = True
    Label2.Visible = True
    Label3.Visible = True
    Label4.Visible = True
    Label5.Visible = True
    Label6.Visible = True
    Label7.Visible = True
    Label8.Visible = True
    Timer1.Enabled = True
End Sub
```

Figure 10: Screenshot of Source Codes for “Begin Vote” Key

The key function of “the result” key was designed to calculate the results of the voting process and could use the following labels:

- Label-4 was used to view the no. of voters that said “Yes”.
- Label-5 was used to view the no. of voters that said “No”.
- Label 8 was used to view the total no. voters.
- Label 2 was used to find out the law (“pass” or “fail”) or retry the voting process again.

All these options can be programmed as shown in figure 11.



```
Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button2.Click
    For i = 0 To 2
        If ListBox2.Items(i) = "yes" Then

            yes = yes + 1
        ElseIf ListBox2.Items(i) = "no" Then
            no = no + 1
        ElseIf ListBox2.Items(i) = "idel" Then
            idel = idel + 1
        End If
    Next i
    Label4.Text = yes
    Label6.Text = no
    Label8.Text = yes + no + idel
    If (yes + no + idel) >= 2 Then
        If (yes > no) Then
            Label2.Text = "the low is pass"
        ElseIf (yes < no) Then
            Label2.Text = "the low is fail"
        ElseIf (yes = no) Then
            Label2.Text = "retry the voting"
        End If
    Else : Label2.Text = "the no. of voters less than the mid"

    End If
    Button2.Enabled = False
End Sub
```

Figure 11: Source Codes for "The Result" Key Function and Labels

The main computer program contained also another key function called ”search by name”, which help the chairman to check the answer status for each voter, figure 12 shows the screenshot for this key function.

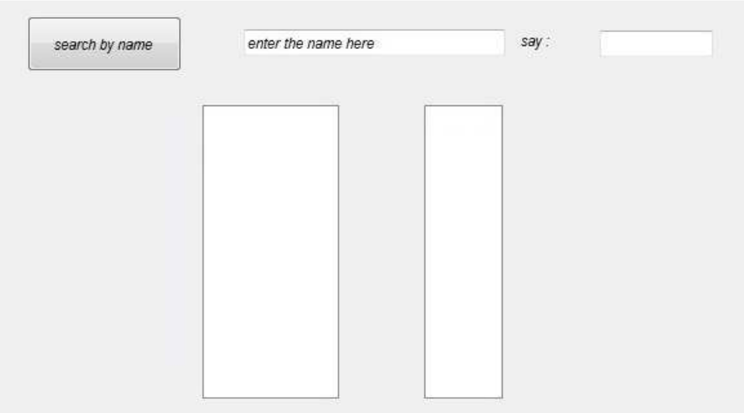
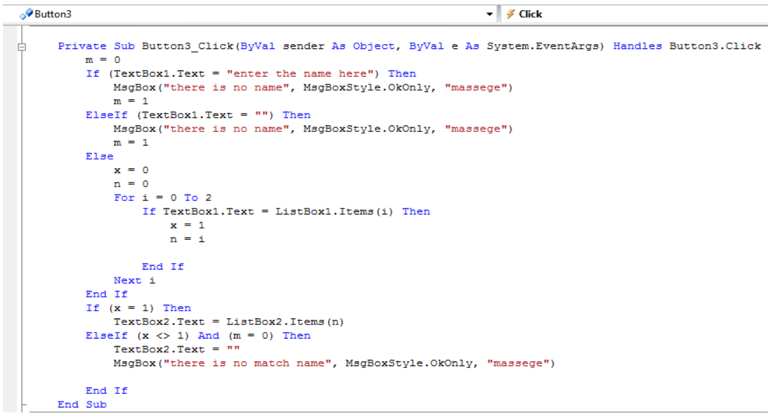


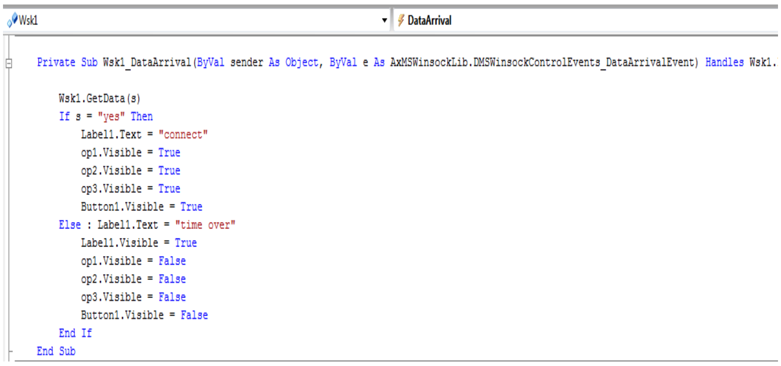
Figure 12: Screen Shot for the "Search by Name" Key Function



```
Private Sub Button3_Click(ByVal sender As Object, ByVal e As System.EventArgs) Handles Button3.Click
    m = 0
    If (TextBox1.Text = "enter the name here") Then
        MsgBox("there is no name", MsgBoxStyle.OkOnly, "massege")
        m = 1
    ElseIf (TextBox1.Text = "") Then
        MsgBox("there is no name", MsgBoxStyle.OkOnly, "massege")
        m = 1
    Else
        x = 0
        n = 0
        For i = 0 To 2
            If TextBox1.Text = ListBox1.Items(i) Then
                x = 1
                n = i

            End If
        Next i
    End If
    If (x = 1) Then
        TextBox2.Text = ListBox2.Items(n)
    ElseIf (x <> 1) And (m = 0) Then
        TextBox2.Text = ""
        MsgBox("there is no match name", MsgBoxStyle.OkOnly, "massege")

    End If
End Sub
```

Figure 13: Source Code for the "Search by Name" Key Function

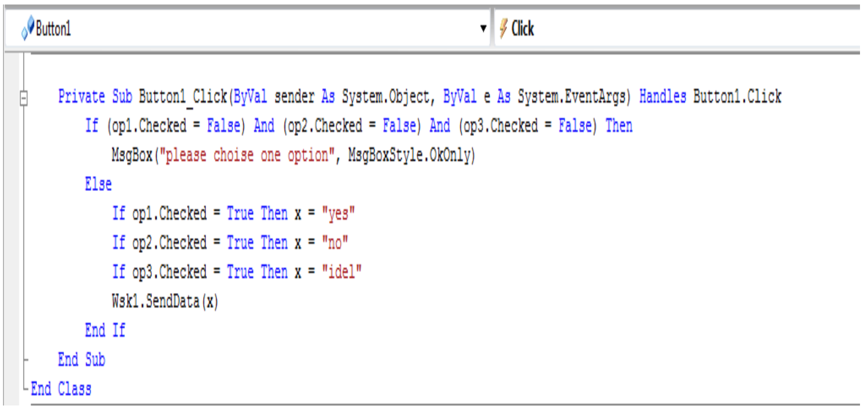
The source codes for data arrival, can be written as shown in figure 14.



```
Private Sub Wsk1_DataArrival(ByVal sender As Object, ByVal e As AxMSWinsockLib.DMSWinsockControlEvents_DataArrivalEvent) Handles Wsk1.

    Wsk1.GetData(s)
    If s = "yes" Then
        Label1.Text = "connect"
        op1.Visible = True
        op2.Visible = True
        op3.Visible = True
        Button1.Visible = True
    Else : Label1.Text = "time over"
        Label1.Visible = True
        op1.Visible = False
        op2.Visible = False
        op3.Visible = False
        Button1.Visible = False
    End If
End Sub
```

Figure 14: Source Code for the Data Arrival

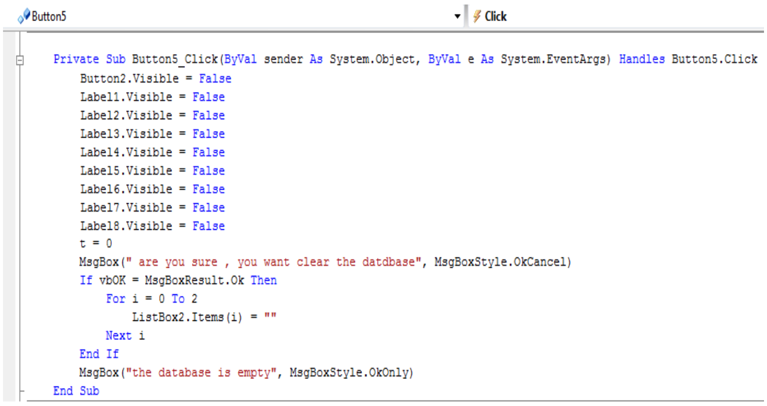
The selection process of voting can be programmed as shown in figure 15.



```
    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
        If (op1.Checked = False) And (op2.Checked = False) And (op3.Checked = False) Then
            MsgBox("please choise one option", MsgBoxStyle.OkOnly)
        Else
            If op1.Checked = True Then x = "yes"
            If op2.Checked = True Then x = "no"
            If op3.Checked = True Then x = "idel"
            Wsk1.SendData(x)
        End If
    End Sub
End Class
```

Figure 15: Source Code for the Selection Voting Process

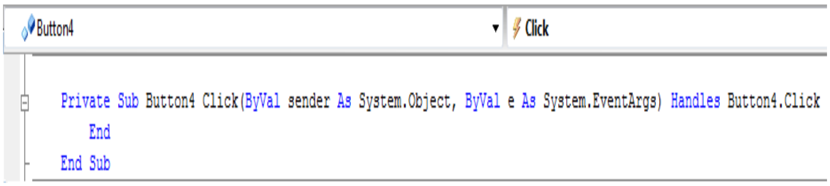
To clear the data base, we use the following source codes as shown in figure 16.



```
Private Sub Button5_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button5.Click
    Button2.Visible = False
    Label1.Visible = False
    Label2.Visible = False
    Label3.Visible = False
    Label4.Visible = False
    Label5.Visible = False
    Label6.Visible = False
    Label7.Visible = False
    Label8.Visible = False
    t = 0
    MsgBox(" are you sure , you want clear the datdbase", MsgBoxStyle.OkCancel)
    If vbOK = MsgBoxResult.Ok Then
        For i = 0 To 2
            ListBox2.Items(i) = ""
        Next i
    End If
    MsgBox("the database is empty", MsgBoxStyle.OkOnly)
End Sub
```

Figure 16: Source Code for Clearing Database

The “exit” key function was programmed by using the source codes shown in figure 17.



```
Private Sub Button4 Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button4.Click
    End
End Sub
```

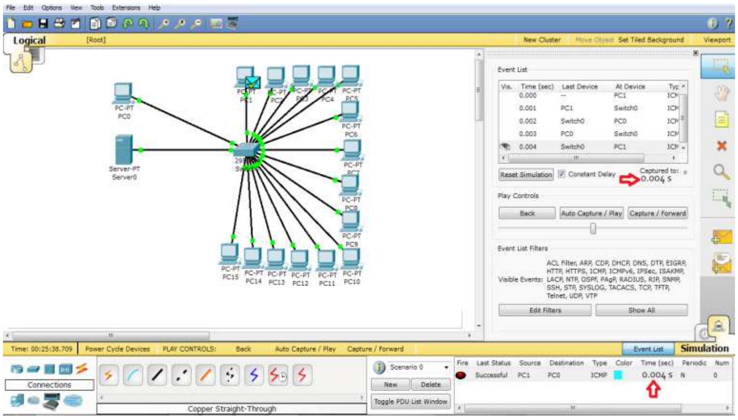
Figure 17: Source Code for “Exit” Key Function

THE IMPLEMENTATION OF A PROPOSAL E-VOTING SYSTEM

In this section we will test and check the conventional and our proposal hardware design of the networked e-voting system, we will also test the capability of the designed software package.

Test for the Hardware Connection in Scenario - 1

The simulated results of the hardware connections for the e-voting system in scenario-1, give the time of (0.004 sec) as a required time for sending the voting message from pc.1 to the main pc., while the time taken for sending voting message from pc.15 to main pc. was (0.008 sec). figure 18, and figure 19 shows the simulated results for the system in scenari-1.



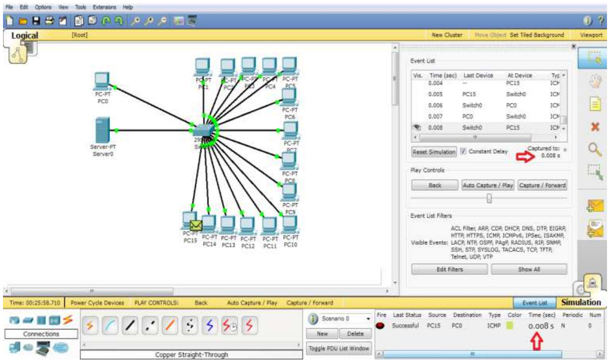
Vis.	Time (sec)	Last Device	At Device	Typ
	0.000	--	PC1	ICM
	0.001	PC1	Switch0	ICM
	0.002	Switch0	PC0	ICM
	0.003	PC0	Switch0	ICM
	0.004	Switch0	PC1	ICM

☑ Constant Delay Captured to: 0.004 s

Visible Events: ACL Filter, ARP, CDP, DHCP, DNS, DTP, EIGRP, HTTP, HTTPS, ICMP, ICMPv6, IPSec, ISAKMP, LACP, NTP, OSPF, PAgP, RADIUS, RIP, SNMP, SSH, STP, SYSLOG, TACACS, TCP, TFTP, Telnet, UDP, VTP

Fire	Last Status	Source	Destination	Type	Color	Time (sec)	Periodic	Num
●	Successful	PC1	PC0	ICMP		0.004 s	N	0

Figure 18: Simulated Results for Sending Voting Message from pc.1 to Main pc. For the System in Scenario-1



Vis.	Time (sec)	Last Device	At Device	Typ
	0.004	--	PC15	ICM
	0.005	PC15	Switch0	ICM
	0.006	Switch0	PC0	ICM
	0.007	PC0	Switch0	ICM
	0.008	Switch0	PC15	ICM

☑ Constant Delay Captured to: 0.008 s

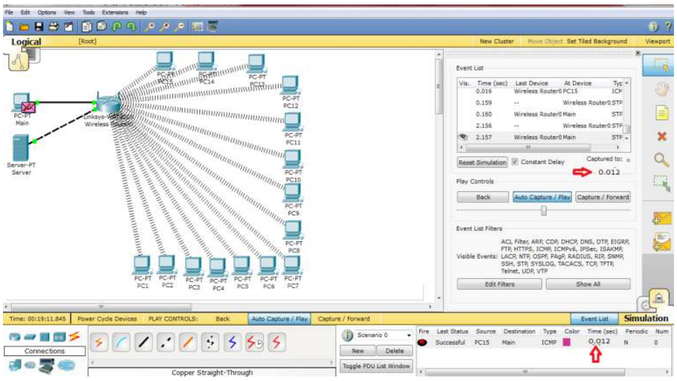
Visible Events: ACL Filter, ARP, CDP, DHCP, DNS, DTP, EIGRP, HTTP, HTTPS, ICMP, ICMPv6, IPSec, ISAKMP, LACP, NTP, OSPF, PAgP, RADIUS, RIP, SNMP, SSH, STP, SYSLOG, TACACS, TCP, TFTP, Telnet, UDP, VTP

Fire	Last Status	Source	Destination	Type	Color	Time (sec)	Periodic	Num
●	Successful	PC15	PC0	ICMP		0.008 s	N	0

Figure 19: Simulated Results for Sending Voting Message from pc.15 to Main pc. For the System in Scenario-1

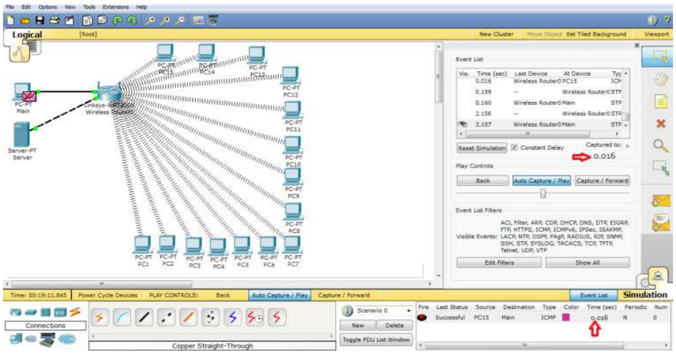
Test for the Hardware Connection in Scenario - 2

The simulated results of the hardware connections for the e-voting system in scenario-2, give time of (0.12 sec) as a required time for sending the voting message from pc.1 to the main pc., while the time taken for sending voting message from pc.15 to main pc. was (0.016 sec), in spite sending voting message in system of scenario-2 took long time duration than that in scenario-1, but we prefer the hardware design of system in scenario-2, because this system is reliable and easy to upgrade, modify, and to increase the number of voting pcs., also we noted that the duration time of sending the voting message through pcs. of this system was within suitable range, figure 20, and figure 21 show the simulated results for the system in scenario-2.



```
Event List
Vis.  Time (sec)  Last Device       At Device         Typ
      0.016       Wireless Router0  PC15              ICM
      0.159       --                Wireless Router0  STP
      0.160       Wireless Router0  Main              STP
      2.156       --                Wireless Router0  STP
      2.157       Wireless Router0  Main              STP
Reset Simulation   ☑ Constant Delay   Captured to:
                                      0.012
Play Controls
Back   Auto Capture / Play   Capture / Forward
Event List Filters
Visible Events: ACL Filter, ARP, CDP, DHCP, DNS, DTP, EIGRP,
                FTP, HTTPS, ICMP, ICMPv6, IPSec, ISAKMP,
                LACP, NTP, OSPF, PAgP, RADIUS, RIP, SNMP,
                SSH, STP, SYSLOG, TACACS, TCP, TFTP,
                Telnet, UDP, VTP
Edit Filters   Show All

Fire  Last Status  Source  Destination  Type  Color  Time (sec)  Periodic  Num
●     Successful   PC15    Main         ICMP  🟪     0.012       N         0
```

Figure 20: Simulated Results for Sending Voting Message from pc.1 to Main pc. For the System in Scenario-2

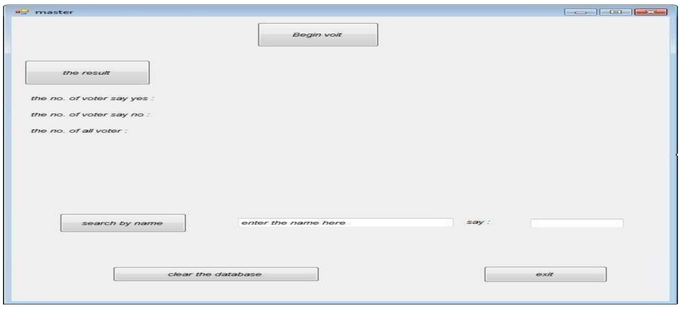


```
Event List
Vis.  Time (sec)  Last Device       At Device         Typ
      0.016       Wireless Router0  PC15              ICM
      0.159       --                Wireless Router0  STP
      0.160       Wireless Router0  Main              STP
      2.156       --                Wireless Router0  STP
      2.157       Wireless Router0  Main              STP
Reset Simulation   ☑ Constant Delay   Captured to:
                                      0.016
Play Controls
Back   Auto Capture / Play   Capture / Forward
Event List Filters
Visible Events: ACL Filter, ARP, CDP, DHCP, DNS, DTP, EIGRP,
                FTP, HTTPS, ICMP, ICMPv6, IPSec, ISAKMP,
                LACP, NTP, OSPF, PAgP, RADIUS, RIP, SNMP,
                SSH, STP, SYSLOG, TACACS, TCP, TFTP,
                Telnet, UDP, VTP
Edit Filters   Show All

Fire  Last Status  Source  Destination  Type  Color  Time (sec)  Periodic  Num
●     Successful   PC15    Main         ICMP  🟪     0.016       N         0
```

Figure 21: Simulated Results for Sending Voting Message from pc.15 to Main pc. For the System in Scenario-2

Test for the Designed Software Package

The executed results of the designed software package for the proposal e-voting system gave the following screenshots.



the no. of voter say yes :

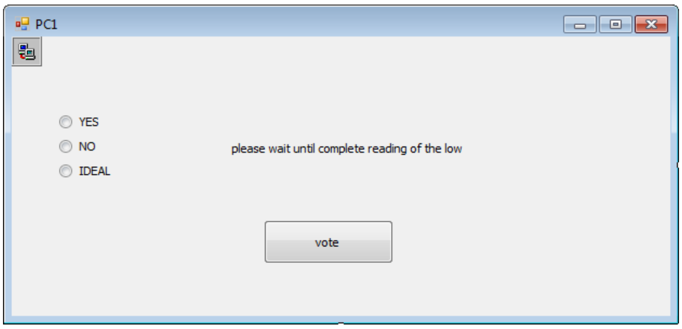
the no. of voter say no :

the no. of all voter :

enter the name here

say :

Figure 22: The Execution Screen for the Program of Main pc



○ YES

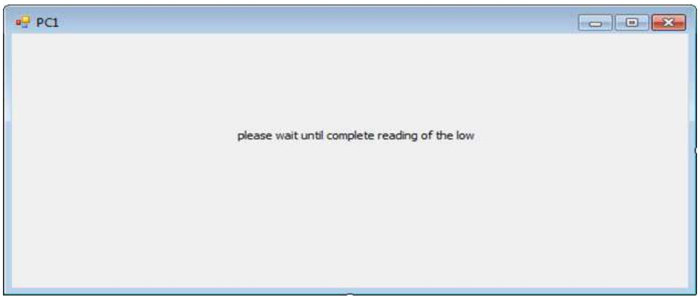
○ NO

○ IDEAL

please wait until complete reading of the low

Figure 23: The Execution Screen for the Program of Voter pc

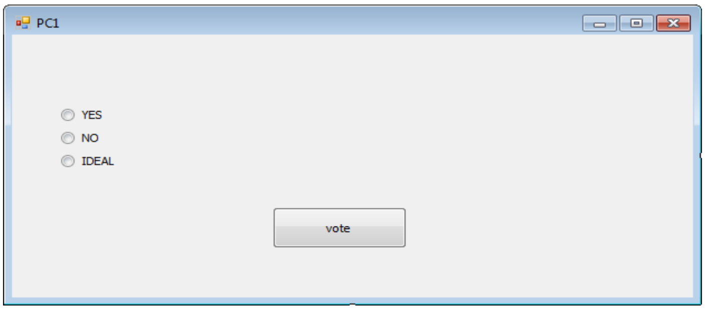
The execution of voter program can give the following results as shown in the below figures.



PC1

please wait until complete reading of the low

Figure 24: The Execution Screen for the Voter Program before Starting the Operation of Voting



PC1

○ YES
○ NO
○ IDEAL

Figure 25: The Execution Screen for the Voter Program when Beginning the Operation of Voting

CONCLUSIONS

From this work, and the practical results we concluded the following points;

- The proposal wireless e-voting system gave the reliability, and flexibility for increasing the number of voters.
- The use of wired connections between main pc. and main server gave more security and authentication for transferring data between them.
- The use of Wi Fi signal in our proposal system gave the flexibility of sending the voted signal through the mobile phones.
- The proposal e-voting system gave suitable results for scoring and storing the voting numbers, as well as storing extra information about the voted law or paragraph.

REFERENCES

1. Chun-Ta Li, Min-Shiang Hwang, and Chi-Yu Liu, “An electronic voting protocol with deniable authentication for mobilead hoc networks”, journal of Computer Communications, Taiwan ROC, vol. 31, pp. 2534–2540, 2008.
2. Ahmed Abdel-Hafez, Ali Miri, and Louis Orozco -Barbosa, “Authenticated group key agreement protocols for Ad hoc wireless networks”, International Journal of Network Security, vol.4, pp. 90–98, 2007.
3. Yonatan Aumann, and Michael Rabin, “Authentication enhanced security and error correcting codes”, in: Proceedings of the 18th Annual International Cryptology Conference on Advances in Cryptology, Lecture Notes in Computer Science, vol. 1462, pp. 299–303, 1998.

4. Yonatan Aumann and Michael Rabin, “Efficient deniable authentication of long messages”, International Conference on Theoretical Computer Science in Honor of Professor Manuel Blum’s 60th Birthday, Hong Kong, China, 1998.

5. F. Baiardi, A. Falleni, R. Granchi, F. Martinelli, M. Petrocchi, and A. Vaccarelli, Seas, “A secure E-voting protocol: design and implementation, Computers & Security”, 8th ed., pp. 642–652, 2005.

6. Azzedine Boukerche, Khalil El-Khatib, Li Xu, andLarry Korba, “An efficient secure distributed anonymous routing protocol for mobile and wireless ad hoc networks”, Computer Communications, vol. 28-(10), pp. 1193–1203, 2005.

7. C.C. Chang, and J.S. Lee, “An anonymous voting mechanism based on the key exchange protocol ”l, Computers & Security, vol. 25, pp. 307–314, 2006.

8. D. Chaum, “Untraceable electronic mail return addresses and digital pseudonyms”, Communications of the ACM, vol. 24, pp.84–88, 1981.

9. Yu-Yi Chen, Jinn-Ke Jan, and Chin-Ling Chen, “The design of a secure anonymous Internet voting system”, Computers & Security, vol. 23, pp. 330–337, 2004.

10. Ghassan Z., Qadah, and Rani Taha, “Electronic voting systems: Requirements, design, and implementation”, Science Direct - Computer Standards & Interfaces, vol. 29, pp. 376-386, 2007.